

Parameterized Shifted Combinatorial Optimization

Jakub Gajarský^{*1}, Petr Hliněný^{**1}, Martin Koutecký^{***2}, and Shmuel Onn^{†3}

¹ Masaryk University, Brno, Czech Republic. Email:
{gajarsky,hlineny}@fi.muni.cz

² Charles University, Prague. Email: koutecky@kam.mff.cuni.cz

³ Technion - Israel Institute of Technology, Haifa, Israel. Email:
onn@ie.technion.ac.il

Abstract. *Shifted combinatorial optimization* is a new nonlinear optimization framework which is a broad extension of standard combinatorial optimization, involving the choice of several feasible solutions at a time. This framework captures well studied and diverse problems ranging from so-called vulnerability problems to sharing and partitioning problems. In particular, every standard combinatorial optimization problem has its shifted counterpart, which is typically much harder. Already with explicitly given input set the shifted problem may be NP-hard. In this article we initiate a study of the parameterized complexity of this framework. First we show that shifting over an explicitly given set with its cardinality as the parameter may be in XP, FPT or P, depending on the objective function. Second, we study the shifted problem over sets definable in MSO logic (which includes, e.g., the well known MSO partitioning problems). Our main results here are that shifted combinatorial optimization over MSO definable sets is in XP with respect to the MSO formula and the treewidth (or more generally clique-width) of the input graph, and is W[1]-hard even under further severe restrictions.

Keywords: combinatorial optimization; shifted problem; treewidth; MSO logic; MSO partitioning

^{*} Current affiliation: Technical University Berlin, e-mail jakub.gajarsky@tu-berlin.de
J. Gajarský's research was partially supported by the European Research Council under the European Union's Horizon 2020 research and innovation programme (ERC Consolidator Grant DISTRUCT, grant agreement No 648527).

^{**} P. Hliněný, and partially J. Gajarský, were supported by the the research centre Institute for Theoretical Computer Science (CE-ITI), project P202/12/G061 of the Czech Science Foundation.

^{***} M. Koutecký was partially supported by the project 17-09142S of the Czech Science Foundation.

[†] Shmuel Onn was partially supported by the Dresner Chair at the Technion.

1 Introduction

The following optimization problem has been studied extensively in the literature.

(Standard) Combinatorial Optimization. Given $S \subseteq \{0, 1\}^n$ and $\mathbf{w} \in \mathbb{Z}^n$, solve

$$\max\{\mathbf{w}\mathbf{s} \mid \mathbf{s} \in S\}. \quad (1)$$

The complexity of the problem depends on \mathbf{w} and the type and presentation of S . Often, S is the set of indicating (characteristic) vectors of members of a family of subsets over a ground set $[n] := \{1, \dots, n\}$, such as the family of $s - t$ dipaths in a digraph with n arcs, the set of perfect matchings in a bipartite or arbitrary graph with n edges, or the set of bases in a matroid over $[n]$ given by an independence oracle.

Partly motivated by vulnerability problems studied recently in the literature (see a brief discussion below), in this article we study a broad nonlinear extension of CO, in which the optimization is over r choices of elements of S and which is defined as follows. For a set $S \subseteq \mathbb{R}^n$, let S^r denote the set of $n \times r$ matrices having each column in S ,

$$S^r := \{\mathbf{x} \in \mathbb{R}^{n \times r} \mid \mathbf{x}^k \in S, k = 1, \dots, r\}.$$

Call $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n \times r}$ equivalent and write $\mathbf{x} \sim \mathbf{y}$ if each row of \mathbf{x} is a permutation of the corresponding row of \mathbf{y} . The *shift* of $\mathbf{x} \in \mathbb{R}^{n \times r}$ is the unique matrix $\bar{\mathbf{x}} \in \mathbb{R}^{n \times r}$ satisfying $\bar{\mathbf{x}} \sim \mathbf{x}$ and $\bar{\mathbf{x}}^1 \geq \dots \geq \bar{\mathbf{x}}^r$, that is, the unique matrix equivalent to \mathbf{x} with each row nonincreasing. Our nonlinear optimization problem follows:

Shifted Combinatorial Optimization (SCO). Given $S \subseteq \{0, 1\}^n$ and $\mathbf{c} \in \mathbb{Z}^{n \times r}$, solve

$$\max\{\mathbf{c}\bar{\mathbf{x}} \mid \mathbf{x} \in S^r\}. \quad (2)$$

(Here $\mathbf{c}\bar{\mathbf{x}}$ is used to denote the ordinary scalar product of the vectors \mathbf{c} and $\bar{\mathbf{x}}$.)

This problem easily captures many classical fundamental problems. For example, given a graph $G = (V, E)$ with n vertices, let $S := \{N[v] \mid v \in V\} \subseteq \{0, 1\}^n$, where $N[v]$ is the characteristic vector of the closed neighborhood of v . Choose an integer parameter r and let $c_i^1 := 1$ for all i and $c_i^j := 0$ for all i and all $j \geq 2$. Then the optimal objective function value of (2) is n if and only if we can select a set D of r vertices in G such that every vertex belongs to the closed neighborhood of at least one of the selected vertices, that is, when D is a dominating set of G . Likewise, one can formulate the vertex cover and independent set problems in a similar way.

One specific motivation for the SCO problem is as follows. Suppose S is the set of indicators of members of a family over $[n]$. A feasible solution $\mathbf{x} \in S^r$ then represents a choice of r members of the given family such that the k -th column \mathbf{x}^k is the indicator of the k -th member. Call element i in the ground set *k-vulnerable* in $\bar{\mathbf{x}}$ if it is used by at least k of the members represented by \mathbf{x} , that is, if the i -th row \mathbf{x}_i of \mathbf{x} has at least k ones. It is easy to see that the k -th

column $\bar{\mathbf{x}}^k$ of the shift of \mathbf{x} is precisely the indicator of the set of k -vulnerable elements in \mathbf{x} . So the shifted optimization problem is to maximize

$$\mathbf{c}\bar{\mathbf{x}} = \sum \{c_i^k \mid i \text{ is } k\text{-vulnerable in } \mathbf{x}, i = 1, \dots, n, k = 1, \dots, r\}.$$

Minimizing the numbers of k -vulnerable elements in \mathbf{x} may be beneficial for survival of some family members under various attacks to vulnerable elements by an adversary, see e.g. [1,21] for more details. For example, to minimize the number of k -vulnerable elements for some k , we set $c_i^k := -1$ for all i and $c_i^j := 0$ for all i and all $j \neq k$. To *lexicographically* minimize the numbers of r -vulnerable elements, then of $(r-1)$ -vulnerable elements, and so on, till that of 1-vulnerable elements, we can set $c_i^k := -(n+1)^{k-1}$ for all i, k .

As another natural example, consider \mathbf{c} with $c_i^1 := 1$ and $c_i^j := -1$ for $1 < j \leq r$. Then $\mathbf{c}\bar{\mathbf{x}} = n$ if and only if the columns of \mathbf{x} indicate a *partition* of S . This formulation hence allows us to optimize over partitions of the ground set (see Section 4). Or, consider \mathbf{c} with $\mathbf{c}_i = (1, \dots, 1, -1, \dots, -1)$ of length $a > 0$ with $b \leq a$ ones, and let S be the family of independent sets of a graph G . Then $\max \mathbf{c}\bar{\mathbf{x}}$ relates to *fractional coloring* of G ; it holds $\max \mathbf{c}\bar{\mathbf{x}} = bn$ if and only if G has a coloring by a colors in total such that every vertex receives b distinct colors – this is the so-called $(a : b)$ -COLORING problem.

The complexity of the shifted combinatorial optimization (SCO) problem depends on \mathbf{c} and on the presentation of S , and is typically harder than the corresponding standard combinatorial optimization problem. Say, when S is the set of perfect matchings in a graph, the standard problem is polynomial time solvable, but the shifted problem is NP-hard even for $r = 2$ and cubic graphs, as the optimal value of the above 2-vulnerability problem is 0 if and only if the graph is 3-edge-colorable [18]. The minimization of 2-vulnerable arcs with S the set of s - t dipaths in a digraph, also called the MINIMUM SHARED EDGES problem, was recently shown to be NP-hard for r variable in [21], polynomial time solvable for fixed r in [1], and fixed-parameter tractable with r as a parameter in [6].

In the rest of this article we always assume that the number r of choices is variable. Call a matrix $\mathbf{c} \in \mathbb{Z}^{n \times r}$ *shifted* if $\mathbf{c} = \bar{\mathbf{c}}$, that is, if its rows are nonincreasing. In [14] it was shown that when $S = \{\mathbf{s} \in \{0, 1\}^n \mid \mathbf{A}\mathbf{s} = \mathbf{b}\}$ where \mathbf{A} is a totally unimodular matrix and \mathbf{b} is an integer vector, the shifted problem with shifted \mathbf{c} , and hence in particular the above lexicographic vulnerability problem, can be solved in polynomial time. In particular this applies to the cases of S the set of s - t dipaths in a digraph and S the set of perfect matchings in a bipartite graph. In [18] it was shown that the shifted problem with shifted \mathbf{c} is also solvable in polynomial time for S the set of bases of a matroid presented by an independence oracle (in particular, spanning trees in a graph), and even for the intersection of matroids of certain type.

Main results and paper organization. In this article we continue on systematic study of shifted combinatorial optimization. The paper is organized as follows. Preliminaries, including necessary basics of parameterized complexity (FPT, XP and W[1]-hardness) and of logic (MSO on graphs), are in Section 2.

Then, in Section 3, we consider the case when the set S is given explicitly. While the standard problem is always trivial in such case, the SCO problem can be NP-hard for explicit set S (Proposition 3.1). Our main results on this case can be briefly summarized as follows:

- **(Theorem 3.2)** The shifted combinatorial optimization problem, parameterized by $|S| = m$, is; (a) for general \mathbf{c} in the complexity class XP and W[1]-hard w.r.t. m , (b) for shifted \mathbf{c} in FPT, and (c) for shifted $-\mathbf{c}$ in P.
- **(Theorem 3.4)** The latter case (c) of shifted $-\mathbf{c}$ is in P even for sets S presented by a linear optimization oracle.

In Section 4, we study a more general framework of SCO for the set S definable in Monadic Second Order (MSO) logic. This rich framework includes, for instance, the well-studied case of so called MSO partitioning problems on graphs. We prove the following statement which generalizes known results about MSO partitioning:

- **(Theorem 4.7, Corollary 4.13)** The shifted combinatorial optimization problem, for (a) graphs of bounded treewidth and S defined in MSO_2 logic, or (b) graphs of bounded clique-width and S defined in MSO_1 logic, is in XP (parameterized by the width and the formula defining S).

In the course of proving this statement we also provide a connection of shifted optimization to separable optimization when the corresponding polyhedron is decomposable and 0/1 (Lemma 4.3).

To complement the previous tractability result, in Section 5 we prove the following negative result under much more restrictive parametrization.

- **(Theorem 5.2)** There exists a fixed First Order formula ϕ such that the associated MSO_1 partitioning problem, and hence also the SCO problem with S defined by ϕ , are W[1]-hard on graphs of bounded treedepth.

We conclude the paper by listing several interesting future research directions in Section 6.

2 Preliminaries

We follow the standard terminology of graph theory and of linear and integer programming. Relevant special terminology is introduced in the corresponding sections of the paper. Here we briefly recall basic terms of parameterized complexity and of logic on graphs.

A *parameterized problem* Q is a subset of $\Sigma^* \times \mathbb{N}_0$, where Σ is a finite alphabet. A parameterized problem Q is said to be *fixed-parameter tractable* if there is an algorithm that given $(x, k) \in \Sigma \times \mathbb{N}_0$ decides whether (x, k) is a YES-instance of Q in time $f(k) \cdot p(|x|)$ where f is some computable function of k alone, p is a polynomial and $|x|$ is the size measure of the input. The class of such problems is denoted by FPT. The class XP is the class of parameterized problems that

admit algorithms with a run-time of $\mathcal{O}(|x|^{f(k)})$ for some computable function f , i.e. polynomial-time for every fixed value of k .

Theory of parameterized complexity, see e.g. [5], defines also complexity classes $W[t]$ for $t \geq 1$, where $W[t] \subseteq XP$ for all integers $t \geq 1$. For instance, the k -independent set problem (with parameter k) is complete for $W[1]$. Problems that are $W[1]$ -hard do not admit an FPT algorithm unless the *Exponential Time Hypothesis* (ETH) fails, which is considered unlikely.

We now shortly introduce *monadic second order logic* (MSO) over graphs. In *first-order logic* (FO) we have variables for the elements (x, y, \dots) , equality for variables, quantifiers \forall, \exists ranging over vertices and vertex sets, and the standard Boolean connectives. MSO is the extension of FO by quantification over sets (X, Y, \dots) . In this context a graph is considered as a *relational structure*; either with only the adjacency relation on its vertex set (i.e., the relational vocabulary consists of one predicate symbol $\text{edge}(x, y)$), or as two-sorted structures with its vertex and edge sets and the incidence relation between those (the vocabulary consists of $\text{inc}(x, e)$, which can also be used to define adjacency).

In correspondence with the previous, graph MSO traditionally comes in two flavours, MSO_1 and MSO_2 , differing by the objects we are allowed to quantify over: in MSO_1 these are the vertices and vertex sets, while in MSO_2 we can additionally quantify over edges and edge sets. For example, the 3-colorability property can be expressed in MSO_1 as follows:

$$\begin{aligned} \exists X_1, X_2, X_3 \quad & [\forall x (x \in X_1 \vee x \in X_2 \vee x \in X_3) \wedge \\ & \bigwedge_{i=1,2,3} \forall x, y (x \notin X_i \vee y \notin X_i \vee \neg \text{edge}(x, y))] \end{aligned}$$

We briefly remark that MSO_2 can express properties which are not MSO_1 definable (e.g., Hamiltonicity).

3 Sets Given Explicitly

In this section we consider the shifted problem (2) over an explicitly given set $S = \{s^1, \dots, s^m\}$. We demonstrate that already this seemingly simple case is in fact nontrivial and interesting. First, notice that with $S \subseteq \{0, 1\}^n$ given explicitly the problem is generally NP-hard, which follows by the reduction from dominating set which we gave in the introduction. Moreover it follows from known lower bounds on the dominating set problem that the brute-force algorithm which tries all possible r -subsets of S is likely close to optimal:

Proposition 3.1. *The SCO problem (2) is NP-hard for 0/1 shifted matrices $c = \bar{c} \in \{0, 1\}^{n \times r}$ and explicitly given 0/1 sets $S = \{s^1, \dots, s^m\} \subseteq \{0, 1\}^n$. Moreover, unless the Exponential Time Hypothesis (ETH) fails, it cannot be solved in time $n^{o(r)}$.*

Proof. The NP-complete *dominating set* problem is to decide whether, given a graph $G = (V, E)$ there is a subset of vertices $D \subseteq V$ of size r such that every

vertex $v \in V$ is either in D , or has a neighbor in D . Let $S := \{N[v] \mid v \in V\} \subseteq \{0, 1\}^n$, where $N[v]$ is the characteristic vector of the closed neighborhood of v , i.e. including v itself, and let $c_i^1 := 1$ for all i and $c_i^j := 0$ for all i and all $j \geq 2$. Then the optimal objective function value of (2) is n if and only if G has a dominating set of size r .

Moreover, Chen et al. [4] proved that unless ETH fails, there is no $n^{o(r)}$ algorithm solving the dominating set problem; thus, under the same assumption, there is no $m^{o(r)}$ algorithm solving SCO even when c is $0/1$ and $c = \bar{c}$. \square

Note that the next results in this section concerning Shifted IP apply to the more general situation in which S may consist of arbitrary integer vectors, not necessarily $0/1$. This is formulated as follows.

Shifted integer programming. Given $S \subseteq \mathbb{Z}^n$ and $\mathbf{c} \in \mathbb{Z}^{n \times r}$, similarly to (2), solve

$$\max\{\mathbf{c}\bar{\mathbf{x}} \mid \mathbf{x} \in S^r\}. \quad (3)$$

For $S = \{\mathbf{s}^1, \dots, \mathbf{s}^m\}$ and nonnegative integers r_1, \dots, r_m with $\sum_{i=1}^m r_i = r$, let $\mathbf{x}(r_1, \dots, r_m)$ be the matrix in S^r with first r_1 columns equal to \mathbf{s}^1 , next r_2 columns equal to \mathbf{s}^2 , and so on, with last r_m columns equal to \mathbf{s}^m , and define $f(r_1, \dots, r_m) := \mathbf{c}\bar{\mathbf{x}}(r_1, \dots, r_m)$.

We have got the following effective theorem in contrast with Proposition 3.1.

Theorem 3.2. *The shifted integer programming problem (3) over an explicitly given set $S = \{\mathbf{s}^1, \dots, \mathbf{s}^m\} \subseteq \mathbb{Z}^n$ reduces to the following nonlinear integer programming problem over a simplex,*

$$\max \left\{ f(r_1, \dots, r_m) \mid r_1, \dots, r_m \in \mathbb{Z}_+, \sum_{k=1}^m r_k = r \right\}. \quad (4)$$

If $\mathbf{c} = \bar{\mathbf{c}}$ is shifted then f is concave, and if $-\mathbf{c}$ is shifted then f is convex.

Moreover, the following hold:

1. *With m parameter and \mathbf{c} arbitrary, problem (3) is in XP. Furthermore, the problem is W[1]-hard with parameter m even for $0/1$ sets S .*
2. *With m parameter and \mathbf{c} shifted, problem (3) is in FPT.*
3. *With m variable and $-\mathbf{c}$ shifted, problem (3) is in P.*

Proof. Consider any $\mathbf{x} \in S^r$. For $k = 1, \dots, m$ let $r_k := |\{j \mid \mathbf{x}^j = \mathbf{s}^k\}|$ be the number of columns of \mathbf{x} equal to \mathbf{s}^k . Then $\mathbf{x} \sim \mathbf{x}(r_1, \dots, r_m)$ so $\bar{\mathbf{x}} = \bar{\mathbf{x}}(r_1, \dots, r_m)$ and $\mathbf{c}\bar{\mathbf{x}} = f(r_1, \dots, r_m)$. So an optimal solution r_1, \dots, r_m to (4) gives an optimal solution $\mathbf{x}(r_1, \dots, r_m)$ to the shifted problem (3), proving the first statement.

We next show that if \mathbf{c} is shifted then f is concave in the r_k . Suppose first that $n = 1$ so that $\mathbf{c}^1, \dots, \mathbf{c}^r$ and $\mathbf{s}^1, \dots, \mathbf{s}^m$ are scalars. For $k = 1, \dots, m$, define functions $g_k(r_1, \dots, r_m) := \sum_{j=1}^k r_j$ which are linear in r_1, \dots, r_m , and define a function h by $h(0) := 0$ and $h(l) := \sum_{j=1}^l \mathbf{c}^j$ for $l = 1, \dots, r$, which is concave since $\mathbf{c}^1 \geq \dots \geq \mathbf{c}^r$.

Let π be a permutation of $[m] = \{1, \dots, m\}$ such that $\mathbf{s}^{\pi(1)} \geq \dots \geq \mathbf{s}^{\pi(m)}$. Consider any r_1, \dots, r_m feasible in (4) and let $\mathbf{x} := \mathbf{x}(r_1, \dots, r_m)$. Note that $\bar{\mathbf{x}}$ is the row vector with first $r_{\pi(1)}$ entries equal to $\mathbf{s}^{\pi(1)}$, next $r_{\pi(2)}$ entries equal to $\mathbf{s}^{\pi(2)}$, and so on, with last $r_{\pi(m)}$ entries equal to $\mathbf{s}^{\pi(m)}$. Let $g_k := g_k(r_{\pi(1)}, \dots, r_{\pi(m)})$ for $k = 1, \dots, m$ and $t^k := \mathbf{s}^{\pi(k)} - \mathbf{s}^{\pi(k+1)} \geq 0$ for $k = 1, \dots, m-1$. Then we have that

$$\begin{aligned} f(r_1, \dots, r_m) &= \mathbf{c}\bar{\mathbf{x}} \\ &= \mathbf{s}^{\pi(1)}h(g_1) + \mathbf{s}^{\pi(2)}(h(g_2) - h(g_1)) + \dots + \mathbf{s}^{\pi(m)}(h(g_m) - h(g_{m-1})) \\ &= \mathbf{t}^1h(g_1) + \mathbf{t}^2h(g_2) + \dots + \mathbf{t}^{m-1}h(g_{m-1}) + \mathbf{s}^{\pi(m)}h(g_m) \\ &= \sum_{k=1}^{m-1} \mathbf{t}^k h(g_k(r_{\pi(1)}, \dots, r_{\pi(m)})) + \mathbf{s}^{\pi(m)} \sum_{j=1}^r \mathbf{c}^j. \end{aligned} \quad (5)$$

Now, g_k are linear functions of r_k , and h is concave, and so each composition $h(g_k(r_{\pi(1)}, \dots, r_{\pi(m)}))$ is also concave. So $f(r_1, \dots, r_m)$, which is a constant plus a nonnegative combination of concave functions, is a concave function of the r_k .

We continue with general n . Consider any r_1, \dots, r_m which are feasible in (4) and let $\mathbf{x} := \mathbf{x}(r_1, \dots, r_m)$. For each $i = 1, \dots, n$ proceed as follows. Let $f_i(r_1, \dots, r_m) := \mathbf{c}_i \bar{\mathbf{x}}_i$ with \mathbf{c}_i the i -th row of \mathbf{c} and $\bar{\mathbf{x}}_i$ the i -th row of the shift $\bar{\mathbf{x}}$. Let π_i be a permutation of $[m]$ such that $\mathbf{s}_i^{\pi_i(1)} \geq \dots \geq \mathbf{s}_i^{\pi_i(m)}$. Repeating the above procedure with this 1-dimensional data we see that $f_i(r_1, \dots, r_m)$ is concave. So $f(r_1, \dots, r_m)$ is also concave in the r_k , being the following sum of concave functions,

$$f(r_1, \dots, r_m) = \mathbf{c}\bar{\mathbf{x}} = \sum_{i=1}^d \mathbf{c}_i \bar{\mathbf{x}}_i = \sum_{i=1}^d f_i(r_1, \dots, r_m).$$

This also shows that if $-\mathbf{c}$ is shifted then $-f$ is concave and hence f is convex.

We proceed with the (positive) algorithmic statements of the theorem. For part 1, which was also proved in [18], just note that for fixed m , there are $\mathcal{O}(r^{m-1})$ feasible solutions in (4), obtained by taking integers $0 \leq r_1, \dots, r_{m-1} \leq r$ with $\sum_{i=1}^{m-1} r_i \leq r$ and setting $r_m := r - \sum_{i=1}^{m-1} r_i$. Hence, in polynomial time we can enumerate all, pick the best, and obtain an optimal solution $\mathbf{x}(r_1, \dots, r_m)$ to the shifted problem (3).

For part 2, if \mathbf{c} is shifted, then, as just shown, f is concave. So the integer program (4) is to maximize a concave function with the number m of variables as a parameter. By known results on convex integer minimization, see [20], this problem is fixed-parameter tractable and solvable in time $p(m)(\log r)^q$ for some computable function p of m and some constant q . Because it is enough to present the objective function by an oracle, we can extend this to the case when \mathbf{c} is not given explicitly, but by a partial sums oracle $\gamma(i, j) = \sum_{\ell=1}^j c_i^\ell$, in which case r can be given in binary.

For part 3, if $-\mathbf{c}$ is shifted, then, as just shown, f is convex. Therefore, the maximum in (4) is attained at a vertex of the simplex. These vertices are

the m vectors $r\mathbf{e}_1, \dots, r\mathbf{e}_m$, where \mathbf{e}_k is the k -th unit vector in \mathbb{R}^m . Hence, in polynomial time we can pick the best vertex $r\mathbf{e}_k$ and obtain again an optimal solution $\mathbf{x}(r\mathbf{e}_k)$ to (3).

To complete the proof of the theorem, we return to the hardness claim in part 1. We proceed by reduction from the MULTIDEMAND SET COVER (MSC) problem, which is as follows. Given is a universe $U = \{u_1, \dots, u_n\}$, a collection of multidemands $\mathbf{d} = (d_1, \dots, d_n)$ where $d_i \subseteq \mathbb{N}$ for $i = 1, \dots, n$, a covering set system $\mathcal{F} = \{F_1, \dots, F_k\} \subseteq 2^U$, and an integer $r \in \mathbb{N}$. The goal is to find an integer partition $r = p_1 + \dots + p_k$ such that, for all $i = 1, \dots, n$, we have $(\sum_{j: u_i \in F_j} p_j) \in d_i$. Knop et al. [15] prove that MSC is W[1]-hard with respect to the parameter n even when $n = k$.

For clarity, we briefly and informally remark on a meaning of the MSC problem. We wish to take each set F_j , $j \in [k]$, with multiplicity p_j , and we demand that for each universe element u_i , $i \in [n]$, the total sum of multiplicities of sets u_i belongs to, falls into the constraint set d_i .

Given an instance $U, \mathbf{d}, \mathcal{F}$ and r of MSC, let $S \subseteq \{0, 1\}^n$ be the set of characteristic vectors of \mathcal{F} , where $|S| = m$ in our case. We will define \mathbf{c} inductively. Fix a row $i \in [n]$ and let $c_i^1 := 1$ if $1 \in d_i$ and $c_i^1 := 0$ otherwise. For $j = 1, \dots, r-1$, let $c_i^{j+1} := 1 - \sum_{\ell=1}^j c_i^\ell$ if $j+1 \in d_i$, and $c_i^{j+1} := -\sum_{\ell=1}^j c_i^\ell$ otherwise. Then $\max_{\mathbf{x} \in S} \mathbf{c}\bar{\mathbf{x}} \leq n$ and $\mathbf{c}\bar{\mathbf{x}} = n$ exactly when the multiplicities p_1, \dots, p_m of the vectors of S in $\mathbf{x} \in S^r$ are such that the number of 1's in each row i of $\bar{\mathbf{x}}$ falls into d_i , by our choice of \mathbf{c} . This is the case if and only if the MSC instance is a “yes” instance. \square

In the rest of this section we provide several supplementary results related to the cases of Theorem 3.2.

Let us first give an exemplary application of part 2 of Theorem 3.2 now. Bredereck et al. [3] study the WEIGHTED SET MULTICOVER (WSM) problem, which is as follows. Given a universe $U = \{u_1, \dots, u_k\}$, integer demands $d_1, \dots, d_k \in \mathbb{N}$ and a multiset $\mathcal{F} = \{F_1, \dots, F_n\} \subseteq 2^U$ with weights $w_1, \dots, w_n \in \mathbb{N}$, find a multiset $\mathcal{F}' \subseteq \mathcal{F}$ of smallest weight which satisfies all the demands – that is, for all $i = 1, \dots, k$, $|\{F \in \mathcal{F}' \mid u_i \in F\}| \geq d_i$. It is shown [3] that this problem is FPT when the size of the universe is a parameter, and then several applications in computational social choice are given there.

Notice that \mathcal{F} can be represented in a succinct way by viewing \mathcal{F} as a set $\mathcal{F}_s = \{F_1, \dots, F_K\}$ and representing the different copies of $F \in \mathcal{F}_s$ in \mathcal{F} by defining K weight functions w_1, \dots, w_K such that, for each $i = 1, \dots, K$, $w_i(j)$ returns the total weight of the first j lightest copies of F_i , or ∞ if there are less than j copies. We call this the *succinct variant*.

Bredereck et al. [3] use Lenstra’s algorithm for their result, which only works when \mathcal{F} is given explicitly. We note in passing that our approach allows us to extend their result to the succinct case.

Proposition 3.3. *Weighted Set Multicover is in FPT with respect to universe size k , even in the succinct variant.*

Proof. Let $\mathbf{d}, \mathcal{F}, \mathbf{w}$ be an instance of WEIGHTED SET MULTICOVER with universe of size k , where k is parameter, and let $K = |\mathcal{F}_s| \leq 2^k$. We will construct an SCO instance with S of size $k + K$ such that solving it will correspond to solving the original WSM problem. Since $\max c\bar{\mathbf{x}}$ with $c = \bar{\mathbf{c}}$ is equivalent to $\min c\bar{\mathbf{x}}$ with $c = -\bar{\mathbf{c}}$, we will define a minimization instance with c non-decreasing.

Let $S = (\{(\mathbf{f}_i, \mathbf{e}_i) \mid F_i \in \mathcal{F}_s\} \cup \{(\mathbf{0}, \mathbf{0})\}) \subseteq \{0, 1\}^{k+K}$ where \mathbf{f}_i is the characteristic vector of the set F_i and \mathbf{e}_i is the i -th unit vector. Let W be the total weight of \mathcal{F} and let $D = \sum d_i$ be the total demand. Then, the first k rows of \mathbf{c} are defined as $c_i^j = -W$ if $j \leq d_i$ and $c_i^j = 0$ otherwise. The corresponding partial sums oracle is $\gamma(i, j) = -jW$ if $j \leq d_i$ and $\gamma(i, j) = -d_iW$ otherwise. The remaining K rows of \mathbf{c} are exactly the weight functions w_1, \dots, w_K , that is, $\gamma(k+i, j) = w_i(j)$, where w_i returns DW whenever it should return ∞ .

Let $r = D$ and solve the SCO given above. A solution is represented by multiplicities r_i for $i = 0, \dots, K$, where r_0 is the multiplicity of the $\mathbf{0}$ vector. We interpret it as a WSM solution straightforwardly: r_i means how many copies of F_i we take to the solution, and we always choose the r_i lightest. Observe that if the objective is at most $-(D-1)W$, it means that the solution “hits” all the $-W$ items in the first k rows, which in turn means all demands are satisfied. On the other hand, the objective is more than $-(D-1)W$ only if some demand was not satisfied. Also, if $\min c\bar{\mathbf{x}} \leq -(D-1)W$, then the solution never hit a DW item in the last K rows, which in turn means that, for each $i = 1, \dots, K$, we have never used more copies of F_i than there actually are. Because the objective decomposes into $-DW + \sum_{i=1}^K w_i(r_i)$, where the first term is a constant and the second is exactly the weight of the solution, we have found the optimum. \square

Theorem 3.2, part 3, can be applied also to sets S presented implicitly by an oracle. A *linear optimization oracle* for $S \subseteq \mathbb{Z}^n$ is one that, queried on $w \in \mathbb{Z}^n$, solves the linear optimization problem $\max\{\mathbf{ws} \mid \mathbf{s} \in S\}$. Namely, the oracle either asserts that the problem is infeasible, or unbounded, or provides an optimal solution. As mentioned before, even for $r = 2$, the shifted problem for perfect matchings is NP-hard, and hence for general \mathbf{c} the shifted problem over S presented by a linear optimization oracle is also hard even for $r = 2$. In contrast, we have the following strengthening.

Theorem 3.4. *The shifted problem (3) with \mathbf{c} nondecreasing, over any set $S \subset \mathbb{Z}^n$ which is presented by a linear optimization oracle, can be solved in polynomial time.*

Proof. Let $\mathbf{w} := \sum_{j=1}^r \mathbf{c}^j$ be the sum of the columns of \mathbf{c} , and query the linear optimization oracle of S on \mathbf{w} . If the oracle asserts that the problem is infeasible, then $S = \emptyset$ hence $S^r = \emptyset$ hence so is the shifted problem. Suppose it asserts that the problem is unbounded. Then for every real number q there is an $\mathbf{s} \in S$ with $\mathbf{ws} \geq q$. Then the matrix $\mathbf{x} := [\mathbf{s}, \dots, \mathbf{s}]$ with all columns equal to \mathbf{s} satisfies $\bar{\mathbf{x}} = \mathbf{x}$ and hence $\mathbf{c}\bar{\mathbf{x}} = \mathbf{c}\mathbf{x} = \sum_{j=1}^r \mathbf{c}^j \mathbf{s} = \mathbf{ws} \geq q$, and therefore the shifted problem is also unbounded.

Suppose then that the oracle returns an optimal solution $\mathbf{s}^* \in S$ and define $\mathbf{x}^* := [\mathbf{s}^*, \dots, \mathbf{s}^*]$ to be the matrix with all columns equal to \mathbf{s}^* . We claim that \mathbf{x}^*

is an optimal solution to the shifted problem. Suppose indirectly \mathbf{x} is a strictly better solution. Let T be the set of columns of \mathbf{x} , that means $T := \{\mathbf{x}^1, \dots, \mathbf{x}^r\} = \{\mathbf{t}^1, \dots, \mathbf{t}^m\}$ for suitable distinct $\mathbf{t}^k \in S$, where $k = 1, \dots, m$ and $m = |T| \leq r$.

Consider the shifted problem over T . By the proof of the algorithmic part 2 of Theorem 3.2, we will have an optimal solution $\mathbf{y} := \mathbf{t}(r\mathbf{e}_k) = \mathbf{t}(0, \dots, 0, r, 0, \dots, 0)$ for some unit vector $\mathbf{e}_k \in \mathbb{R}^m$, that is, $\mathbf{y} = [\mathbf{t}, \dots, \mathbf{t}]$ for some $\mathbf{t} \in T$. We then obtain

$$\mathbf{w}\mathbf{t} = \mathbf{c}\mathbf{y} = \mathbf{c}\bar{\mathbf{y}} \geq \mathbf{c}\bar{\mathbf{x}} > \mathbf{c}\bar{\mathbf{x}}^* = \mathbf{c}\mathbf{x}^* = \mathbf{w}\mathbf{s}^*$$

which is a contradiction to the assumed optimality of \mathbf{s}^* , completing the proof. \square

4 MSO-definable Sets: XP for Bounded Treewidth

In this section we study another tractable and rich case of shifted combinatorial optimization, namely that of the set S defined in the MSO logic of graphs. This case, in particular, includes well studied MSO-partitioning framework of graphs (see below) which is tractable on graphs of bounded treewidth and clique-width. In the course of proving our results, it is useful to study a geometric connection of 0/1 SCO problems to separable optimization over decomposable polyhedra.

4.1 Relating SCO to decomposable polyhedra

The purpose of this subsection is to demonstrate how shifted optimization over 0/1 polytopes closely relates to an established concept of decomposable polyhedra. We refer to the book of Ziegler [26] for definitions and terminology regarding polytopes.

Definition 4.1 (Decomposable polyhedron and Decomposition oracle).

A polyhedron $P \subseteq \mathbb{R}^n$ is decomposable if for every $k \in \mathbb{N}$ and every $\mathbf{x} \in kP \cap \mathbb{Z}^n$, there are $\mathbf{x}^1, \dots, \mathbf{x}^k \in P \cap \mathbb{Z}^n$ with $\mathbf{x} = \mathbf{x}^1 + \dots + \mathbf{x}^k$, where $kP = \{k\mathbf{y} \mid \mathbf{y} \in P\}$.

A decomposition oracle for a decomposable P is one that, queried on $k \in \mathbb{N}$ given in unary and on $\mathbf{x} \in kP \cap \mathbb{Z}^n$, returns $\mathbf{x}^1, \dots, \mathbf{x}^k \in P \cap \mathbb{Z}^n$ with $\mathbf{x} = \mathbf{x}^1 + \dots + \mathbf{x}^k$.

This property is also called *integer decomposition property* or being *integrally closed* in the literature. The best known example are polyhedra given by totally unimodular matrices [2]. Furthermore, we will use the following notion.

Definition 4.2 (Integer separable (convex) minimization oracle).

Let $P \subseteq \mathbb{R}^n$ and let $f(\mathbf{x}) = \sum_{i=1}^n f_i(x_i)$ be a separable function on \mathbb{R}^n . An integer separable minimization oracle for P is one that, queried on this f , either reports that $P \cap \mathbb{Z}^n$ is empty, or that it is unbounded, or returns a point $\mathbf{x} \in P \cap \mathbb{Z}^n$ which minimizes $f(\mathbf{x})$.

An integer separable convex minimization oracle for P is an integer separable minimization oracle for P which can only be queried on functions f as above with all f_i convex.

We now formulate how these notions naturally connect with shifted optimization in the case when S is 0/1.

Lemma 4.3. *Let (S, \mathbf{c}, r) be an instance of shifted combinatorial optimization, with $S \subseteq \{0, 1\}^n$, $r \in \mathbb{N}$ and $\mathbf{c} \in \mathbb{Z}^{n \times r}$. Let $P \subseteq [0, 1]^n$ be a polytope such that $S = P \cap \{0, 1\}^n$ and let $Q \subseteq [0, 1]^{n+n'}$ be some extension of P , that is, $P = \{\mathbf{x} \mid (\mathbf{x}, \mathbf{y}) \in Q\}$.*

Then, provided a decomposition oracle for Q and an integer separable minimization oracle for rQ , the shifted problem given by (S, \mathbf{c}, r) can be solved with one call to the optimization oracle and one call to the decomposition oracle. Furthermore, if \mathbf{c} is shifted, an integer separable convex minimization oracle suffices.

To demonstrate Lemma 4.3 we use it to give an alternative proof of the result of Kaibel et al. [14] that the shifted problem is polynomial when $S = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \in \{0, 1\}^n\}$ and \mathbf{A} is totally unimodular. It is known that $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{1}\}$ is decomposable and a decomposition oracle is realizable in polynomial time [24]. Moreover, it is known that an integer separable convex minimization oracle for rP is realizable in polynomial time [13]. Lemma 4.3 implies that the shifted problem is polynomial for this S when \mathbf{c} is shifted.

The reason we have formulated Lemma 4.3 for S given by an extension Q of the polytope P corresponding to S , is the following: while P itself might not be decomposable, there always exists an extension of it which is decomposable.

Proposition 4.4. *Let $P \subseteq \mathbb{R}^n$ be a 0/1 polytope with m vertices. Then it has a decomposable extension $Q \subseteq \mathbb{R}^{n+m}$.*

Proof. Let $V = \{\mathbf{v}_1, \dots, \mathbf{v}_m\} \subseteq \{0, 1\}^n$ be the vertices of P . Then P can be obtained by projecting the m -dimensional simplex:

$$\begin{aligned} \lambda_1 + \dots + \lambda_m &= 1 \\ \lambda_1 \mathbf{v}_1 + \dots + \lambda_m \mathbf{v}_m &= \mathbf{x} \\ \lambda_i &\geq 0 \quad \forall i \in \{1, \dots, m\} \end{aligned}$$

Let Q be the polytope defined by the system above and fix $k \in \mathbb{N}$. Consider an integer point $(\mathbf{x}, \lambda) \in kQ$. Then $\sum_{i=1}^m \lambda_i (\mathbf{v}_i, \mathbf{e}_i)$ (with \mathbf{e}_i the i -th unit vector) is a decomposition of (\mathbf{x}, λ) into vertices of Q , certifying that Q is decomposable. \square

Other potential candidates where Lemma 4.3 could be applied are classes of polytopes that are either decomposable, or allow efficient integer separable (convex) minimization. Some known decomposable polyhedra are stable set polytopes of perfect graphs, polyhedra defined by k -balanced matrices [25], polyhedra defined by nearly totally unimodular matrices [11], etc. Some known cases where integer separable convex minimization is polynomial are for $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \in \{0, 1, \dots, r\}^n\}$ where the Graver basis of \mathbf{A} has small size or when \mathbf{A} is highly structured, namely when \mathbf{A} is either an n -fold product, a transpose of it, or a 4-block n -fold product; see the books of Onn [22] and De Loera, Hemmecke and Köppe [19].

Now we return to the proof of Lemma 4.3.

Proof of Lemma 4.3: Fix $(\mathbf{x}, \mathbf{y}) \in rQ \cap \mathbb{Z}^{n+n'}$ and consider the set

$$Q_{(\mathbf{x}, \mathbf{y})}^r := \left\{ (\mathbf{a}, \mathbf{b}) = ((\mathbf{a}^i, \mathbf{b}^i))_{i=1}^r \mid \sum_{i=1}^r (\mathbf{a}^i, \mathbf{b}^i) = (\mathbf{x}, \mathbf{y}), (\mathbf{a}^i, \mathbf{b}^i) \in Q \cap \{0, 1\}^{n+n'}, 1 \leq i \leq r \right\} \\ \subseteq \left(Q \cap \{0, 1\}^{n+n'} \right)^r$$

Note that $\mathbf{c}\bar{\mathbf{a}}$ is the same for all $(\mathbf{a}, \mathbf{b}) \in Q_{(\mathbf{x}, \mathbf{y})}^r$, and observe that $(Q \cap \{0, 1\}^{n+n'})^r = \bigcup_{(\mathbf{x}, \mathbf{y}) \in (rQ \cap \{0, 1\}^{n+n'})} Q_{(\mathbf{x}, \mathbf{y})}^r$.

Consider now the objective function $\mathbf{c} \in \mathbb{Z}^{n \times r}$. Define $w_i : \{0, \dots, r\} \rightarrow \mathbb{Z}$ for $i = 1, \dots, n$ by $w_i(k) := \sum_{j=1}^k c_i^j$. Note that if \mathbf{c} is shifted, every w_i is concave as it is a partial sum of a non-increasing sequence. Observe that $\mathbf{c}\bar{\mathbf{a}} = \sum_{i=1}^n w_i(\sum_{j=1}^r a_i^j) = \sum_{i=1}^n w_i(x_i)$. It follows that the minimum of $\sum_{i=1}^n w_i(x_i)$ over $rQ \cap \mathbb{Z}^{n+n'}$ equals the minimum of $\mathbf{c}\bar{\mathbf{a}}$ over $(Q \cap \{0, 1\}^{n+n'})^r$.

To solve the shifted problem, let $f = \sum_{i=1}^{n+n'} f_i$ with $f_i := w_i$ for $1 \leq i \leq n$ and $f_i := 0$ for $n+1 \leq i \leq n+n'$, and query the integer separable minimization oracle on rQ with $-f$ (minimizing $-f$ maximizes f). The oracle returns that the problem is either infeasible or unbounded or returns $(\mathbf{x}, \mathbf{y}) \in rQ$ maximizing f . Next, query the decomposition oracle for Q on r and (\mathbf{x}, \mathbf{y}) to obtain $((\mathbf{x}^i, \mathbf{y}^i))_{i=1}^r$, and return this as the solution. If \mathbf{c} is shifted then $-f$ is convex and an integer separable convex minimization oracle suffices for the first step. \square

4.2 XP algorithm for MSO-definable set

We start with defining the necessary specialized terms.

Definition 4.5 (Treewidth). *Given a graph G , a tree-decomposition of G is an ordered pair (T, \mathcal{W}) , where T is a tree and $\mathcal{W} = \{W_x \subseteq V(G) \mid x \in V(T)\}$ is a collection of bags (vertex sets of G), such that the following hold:*

1. $\bigcup_{x \in V(T)} W_x = V(G)$;
2. for every edge $e = uv$ in G , there exists $x \in V(T)$ such that $u, v \in W_x$;
3. for each $u \in V(G)$, the set $\{x \in V(T) \mid u \in W_x\}$ induces a subtree of T .

The width of a tree-decomposition (T, \mathcal{W}) is $(\max_{x \in V(T)} |W_x|) - 1$. The treewidth of G , denoted $\text{tw}(G)$, is the smallest width of a tree-decomposition of G .

Given a matrix $\mathbf{A} \in \mathbb{Z}^{n \times m}$, we define the corresponding *Gaifman graph* $G = G(\mathbf{A})$ as follows. Let $V(G) = [m]$. We let $\{i, j\} \in E(G)$ if and only if there is an $r \in [n]$ with $\mathbf{A}[r, i] \neq 0$ and $\mathbf{A}[r, j] \neq 0$. Intuitively, two vertices of G are adjacent if the corresponding variables x_i, x_j occur together in some constraint of $\mathbf{A}\mathbf{x} \leq \mathbf{b}$. The (Gaifman) *treewidth* of a matrix \mathbf{A} is then the treewidth of its Gaifman graph, i.e., $\text{tw}(\mathbf{A}) := \text{tw}(G(\mathbf{A}))$.

The aforementioned MSO-partitioning framework of graphs comes as follows.

MSO-partitioning problem. Given a graph G , an MSO_2 formula φ with one free vertex-set variable and an integer r , the task is as follows;

- to find a partition $U_1 \dot{\cup} U_2 \dots \dot{\cup} U_r = V(G)$ of the vertices of G such that $G \models \varphi(U_i)$ for all $i = 1, \dots, r$, or
- to confirm that no such partition of $V(G)$ exists.

For example, if $\varphi(X)$ expresses that X is an independent set, then the φ -MSO-PARTITIONING problem decides if G has an r -coloring, and thus, finding minimum feasible r (simply by trying $r = 1, 2, \dots$) solves the CHROMATIC NUMBER problem. Similarly, if $G \models \varphi(X)$ when X is a dominating set, minimizing r solves the DOMATIC NUMBER problem, and so on.

Rao [23] showed an algorithm for MSO-PARTITIONING, for any MSO₂ formula φ , on a graph G with treewidth $tw(G) = \tau$ running in time $r^{f(\varphi, \tau)} n$ (XP) for some computable function f . Our next result widely generalizes this to SCO over MSO-definable sets.

Definition 4.6 (MSO-definable sets). *For a graph G on $|V(G)| = n$ vertices, we interpret a 0/1 vector $\mathbf{x} \in \{0, 1\}^n$ as the set $X \subseteq V$ where $v \in X$ iff $x_v = 1$. We then say that \mathbf{x} satisfies a formula φ if $G \models \varphi(X)$. Let*

$$S_\varphi(G) = \{\mathbf{x} \mid \mathbf{x} \text{ satisfies } \varphi \text{ in } G\}.$$

Let \mathbf{c} be defined as $c_i^1 := 1$ for $1 \leq i \leq n$ and $c_i^j := -1$ for $2 \leq j \leq r$ and $1 \leq i \leq n$. Observe then the following: deciding whether the shifted problem with $S = S_\varphi(G)$, \mathbf{c} and r , has an optimum of value n is equivalent to solving the MSO-PARTITIONING problem for φ .

Theorem 4.7. *Let G be a graph of treewidth $tw(G) = \tau$, let φ be an MSO₂ formula and $S_\varphi(G) = \{\mathbf{x} \mid \mathbf{x} \text{ satisfies } \varphi\}$. There is an algorithm solving the shifted problem with $S = S_\varphi(G)$ and any given \mathbf{c} and r in time $r^{f(\varphi, \tau)} \cdot |V(G)|$ for some computable function f . In other words, for parameters φ and τ , the problem is in the complexity class XP.*

We will prove Theorem 4.7 using Lemma 4.3 on separable optimization over decomposable polyhedra. To that end we need to show the following two steps:

1. There is an extension Q of the polytope $P = \text{conv}(S_\varphi(G))$ which is decomposable and endowed with a decomposition oracle (Definition 4.1), and
2. there is an integer separable minimization oracle (Definition 4.2) for the polytope rQ .

The first point is implied by a recent result of Kolman, Koutecký and Tiwary [16]:

Proposition 4.8 ([16]). *Let G be a graph on n vertices of treewidth $tw(G) = \tau$, and φ be an MSO₂ formula with one free vertex-set variable. Then, for some computable functions f_1, f_2, f_3 , there are matrices \mathbf{A}, \mathbf{B} and a vector \mathbf{b} , computable in time $f_1(\varphi, \tau) \cdot n$, such that*

1. *the polytope $Q = \{(\mathbf{x}, \mathbf{y}) \mid \mathbf{Ax} + \mathbf{By} = \mathbf{b}, \mathbf{y} \geq \mathbf{0}\} \subseteq [0, 1]^{f_2(\varphi, \tau)n}$ is an extension of the polytope $P = \text{conv}(S_\varphi(G))$,*
2. *Q is decomposable and endowed with a decomposition oracle, and*

3. the (Gaifman) treewidth of the matrix $(\mathbf{A}\mathbf{B})$ is at most $f_3(\varphi, \tau)$.

The second requirement of Lemma 4.3 follows from efficient solvability of the constraint satisfaction problem (CSP) of bounded treewidth, originally proven by Freuder [8]. We will use a natural weighted version of this folklore result.

Definition 4.9 (CSP). An instance $I = (V, \mathcal{D}, \mathcal{H}, \mathcal{C})$ of CSP consists of

- a set of variables z_v , one for each $v \in V$; without loss of generality we assume that $V = \{1, \dots, n\}$,
- a set \mathcal{D} of finite domains $D_v \subseteq \mathbb{Z}$ (also denoted $D(v)$), one for each $v \in V$,
- a set of hard constraints $\mathcal{H} \subseteq \{C_U \mid U \subseteq V\}$ where each hard constraint $C_U \in \mathcal{H}$ with $U = \{i_1, i_2, \dots, i_k\}$ and $i_1 < \dots < i_k$, is a $|U|$ -ary relation $C_U \subseteq D_{i_1} \times D_{i_2} \times \dots \times D_{i_k}$,
- a set of weighted soft constraints $\mathcal{C} \subseteq \{w_U \mid U \subseteq V\}$ where each $w_U \in \mathcal{C}$ with $U = \{i_1, i_2, \dots, i_k\}$ and $i_1 < \dots < i_k$ is a function $w_U : D_{i_1} \times D_{i_2} \times \dots \times D_{i_k} \rightarrow \mathbb{R}$.

For a vector $z = (z_1, z_2, \dots, z_n)$ and a set $U = \{i_1, i_2, \dots, i_k\} \subseteq V$ with $i_1 < i_2 < \dots < i_k$, we define the projection of z on U as $z|_U = (z_{i_1}, z_{i_2}, \dots, z_{i_k})$. A vector $z \in \mathbb{Z}^n$ satisfies the hard constraint $C_U \in \mathcal{H}$ if and only if $z|_U \in C_U$. We say that a vector $z^* = (z_1^*, \dots, z_n^*)$ is a feasible assignment for I if $z^* \in D_1 \times D_2 \times \dots \times D_n$ and z^* satisfies every hard constraint $C \in \mathcal{H}$. The weight of z^* is $w(z^*) = \sum_{w_U \in \mathcal{C}} w_U(z^*|_U)$.

For a CSP instance $I = (V, \mathcal{D}, \mathcal{H}, \mathcal{C})$ one can define the *constraint graph* of I as $G = (V, E)$ where $E = \{\{u, v\} \mid (\exists C_U \in \mathcal{H}) \vee (\exists w_U \in \mathcal{C}) \text{ s.t. } \{u, v\} \subseteq U\}$. The *treewidth* of a CSP instance I is defined as the treewidth of the constraint graph of I .

Proposition 4.10 ([8]). Given a CSP instance I of treewidth τ and maximum domain size $D = \max_{u \in V} |D_u|$, a minimum weight solution can be found in time $\mathcal{O}(D^\tau(n + |\mathcal{H}| + |\mathcal{C}|))$.

Proposition 4.10 can be used to realize an integer separable minimization oracle for integer programs of bounded treewidth, as follows.

Lemma 4.11. Let $\mathbf{A} \in \mathbb{Z}^{n \times m}$, $\mathbf{b} \in \mathbb{Z}^m$, $\boldsymbol{\ell}, \mathbf{u} \in \mathbb{Z}^n$ be given s.t. $\text{tw}(\mathbf{A}) = \tau$, and let $D = \|\mathbf{u} - \boldsymbol{\ell}\|_\infty$. Then an integer separable minimization oracle over $P = \{\mathbf{x} \mid \mathbf{A}\mathbf{x} = \mathbf{b}, \boldsymbol{\ell} \leq \mathbf{x} \leq \mathbf{u}\}$ is realizable in time $D^\tau(n + m)$.

Proof. The proof proceeds by constructing a CSP instance I based on the ILP $\mathbf{A}\mathbf{x} = \mathbf{b}, \boldsymbol{\ell} \leq \mathbf{x} \leq \mathbf{u}$, such that solving I corresponds to integer separable minimization over P . Since the treewidth of I is τ and the maximum domain size is D , Proposition 4.10 does the job.

First, let $V = \{x_1, \dots, x_n\}$. Then, for every $i = 1, \dots, n$, let $D_i = \{\ell_i, \ell_i + 1, \dots, u_i\}$ and $\mathcal{D} = \{D_i \mid i = 1, \dots, n\}$. Observe that $\max_i |D_i| = \|\mathbf{u} - \boldsymbol{\ell}\|_\infty = D$. Regarding hard constraints \mathcal{H} , observe that every row \mathbf{a}_j of \mathbf{A} contains at most $\tau + 1$ non-zeros, since otherwise the Gaifman graph of \mathbf{A} would contain a clique of

size $\tau+2$, contradicting its treewidth of τ . Let $U_j = \{i_1, \dots, i_k\}$, where $k \leq \tau+1$, be the set of indices of non-zero elements of \mathbf{a}_j , and let $x_c = 0$ for all $c \notin U_j$. Let C_{U_j} be the set of assignments from $D_{i_1} \times \dots \times D_{i_k}$ to x_{i_1}, \dots, x_{i_k} that satisfy $\mathbf{a}_j \mathbf{x} = b_j$; obviously $|C_{U_j}| \leq D^k$ and it can be constructed in time $\mathcal{O}(D^k)$. Then, $\mathcal{H} = \{C_{U_j} \mid j = 1, \dots, m\}$. Finally, for a given separable function f such that $f(\mathbf{x}) = \sum_{i=1}^n f_i(x_i)$, let $\mathcal{C} = \{w_{\{x_i\}} \mid i = 1, \dots, n\}$ where $w_{\{x_i\}} = f_i$ for all i .

It is easy to verify that the feasible assignments of I correspond to integer solutions of $\mathbf{A}\mathbf{x} = \mathbf{b}, \ell \leq \mathbf{x} \leq \mathbf{u}$, that its maximum domain size is D and its weight function w is exactly f . Finally, the treewidth of I is τ , since the Gaifman graph of $G(I)$ of I is exactly $G(\mathbf{A})$. Then Proposition 4.10 solves I in time $\mathcal{O}(D^\tau(n + |\mathcal{H}| + |\mathcal{C}|)) = \mathcal{O}(D^\tau(n + m))$, concluding the proof. \square

Consequently, we can finish the proof of Theorem 4.7 by using Lemma 4.3.

Proof of Theorem 4.7: By Proposition 4.8 there is a computable function f such that there exists a polytope $Q = \{(\mathbf{x}, \mathbf{y}) \mid \mathbf{A}\mathbf{x} + B\mathbf{y} = \mathbf{b}, \mathbf{y} \geq \mathbf{0}\} \subseteq [0, 1]^{f_2(\varphi, \tau)n}$ which is an extension of $P_\varphi(G)$; let $F := f_3(\varphi, \tau)$. Also, $S_\varphi(G) = P_\varphi(G) \cap \{0, 1\}^n$. Moreover, Q is decomposable, a decomposition oracle for Q is realizable in polynomial time, and the treewidth of the matrix $(\mathbf{A} \ B)$ is at most F . Note that rQ is given by $\mathbf{A}\mathbf{x} + B\mathbf{y} = r\mathbf{b}, \mathbf{0} \leq (\mathbf{x}, \mathbf{y}) \leq (r, \dots, r)$, so its treewidth is F as well. Lemma 4.11 realizes an integer separable minimization oracle for rQ in time $r^F(n + Fn)$. Since all conditions of Lemma 4.3 are met, this concludes the proof. \square

Besides treewidth, another useful width measure of graphs is the *clique-width* of a graph G . Rao's result [23] applies also to the MSO-partitioning problem for MSO_1 formulas and graphs of bounded clique-width. We show the analogous extension of Theorem 4.7 next.

Definition 4.12 (Clique-width). *This is defined for a graph G as the smallest number of labels $k = cw(G)$ such that some labeling of G can be constructed by an algebraic k -expression using the following operations (where $1 \leq i, j \leq k$):*

1. create a new vertex with label i ;
2. take the disjoint union of two labeled graphs;
3. add all edges between vertices of label i and label j ; and
4. relabel all vertices with label i to have label j .

Corollary 4.13. *Let G be a graph of clique-width $cw(G) = \gamma$ given along with a γ -expression (cf. Definition 4.12), let ψ be an MSO_1 formula and $S_\psi(G) = \{\mathbf{x} \mid \mathbf{x} \text{ satisfies } \psi\}$. There is an algorithm solving the shifted problem with $S = S_\psi(G)$ and any given \mathbf{c} and r in time $r^{f(\psi, \gamma)} \cdot |V(G)|$ for some computable f .*

While it is possible to prove Corollary 4.13 along the same lines as used above, we avoid repeating the previous arguments and, instead, apply the following technical tool. This tool simply extends a folklore fact that a class of graphs is of bounded clique-width if and only if it has an MSO_1 interpretation in the class of rooted trees.

Lemma 4.14. *Let G be a graph of clique-width $cw(G) = \gamma$ given along with a γ -expression Γ constructing G , and let ψ be an MSO_1 formula. One can, in time $\mathcal{O}(|V(G)| + |\Gamma| + |\psi|)$, compute a tree T and an MSO_1 formula φ such that $V(G) \subseteq V(T)$ and*

$$\text{for every } X \text{ it is } T \models \varphi(X), \text{ iff } X \subseteq V(G) \text{ and } G \models \psi(X).$$

Addition to Lemma 4.14. Before giving a (short) proof of this lemma, we need to formally introduce a simplified concept of MSO interpretability. Let σ and ϱ be two relational vocabularies. A *one-dimensional MSO interpretation* of ϱ in σ is a tuple $I = (\nu(x), \{\eta_R(\bar{x})\}_{R \in \varrho})$ of $\text{MSO}[\sigma]$ -formulas where ν has one free element variable and the number of free element variables in η_R is equal to the arity of R in ϱ .

- To every σ -structure A the interpretation I assigns a ϱ -structure A^I with the domain $A^I = \{a \mid A \models \nu(a)\}$ and the relations $R^I = \{\bar{a} \mid A \models \eta_R(\bar{a})\}$ for each $R \in \varrho$. We say that a class \mathcal{C} of ϱ -structures has an interpretation in a class \mathcal{D} of σ -structures if there exists an interpretation I such that for each $C \in \mathcal{C}$ there exists $D \in \mathcal{D}$ such that $C \simeq D^I$, and for every $D \in \mathcal{D}$ the structure D^I is isomorphic to a member of \mathcal{C} .
- The interpretation I of ϱ in σ defines a translation of every $\text{MSO}[\varrho]$ -formula ψ to an $\text{MSO}[\sigma]$ -formula ψ^I as follows:
 - every $\exists x.\phi$ is replaced by $\exists x.(\nu(x) \wedge \phi^I)$,
 - every $\exists X.\phi$ is replaced by $\exists X.(\forall y(y \in X \rightarrow \nu(y)) \wedge \phi^I)$, and
 - every occurrence of a σ -atom $R(\bar{x})$ is replaced by the corresponding formula $\eta_R(\bar{x})$.

It is a folklore fact that for all $\text{MSO}[\varrho]$ -formulas ψ and all σ -structures A

$$A \models \psi^I \iff A^I \models \psi.$$

Let G be a graph of clique-width $cw(G) = \gamma$ given along with a γ -expression Γ constructing G , and let ψ be an MSO_1 formula. One can, in time $\mathcal{O}(|V(G)| + |\Gamma| + |\psi|)$, compute a tree T and an MSO_1 formula φ such that $V(G) \subseteq V(T)$ and

$$\text{for every } X \text{ it is } T \models \varphi(X), \text{ iff } X \subseteq V(G) \text{ and } G \models \psi(X).$$

Proof of Lemma 4.14: Let T_0 be the parse tree of the given γ -expression Γ constructing G . Hence T_0 is a rooted tree such that the set of its leaves is $V(G)$. It is well known that there is a one-dimensional MSO_1 interpretation $I_1 = (\nu_1(x), \eta_1(x, y))$ of the graphs of clique-width $\leq \gamma$ in the class of colored rooted trees (such that the finite set of colors depends only on γ). The used colors are in fact the vertex labels (in the leaves) and the operator symbols (in the internal nodes) from Definition 4.12, and the interpretation I_1 is easy to construct for given γ . See, e.g., [17, Section 4.3] for close details. Consequently, $G \simeq T_0^{I_1}$.

To finish the proof, we just need to “remove” the colors from T_0 and “forget” the root (to make an ordinary uncolored tree T). We first give the root of T_0 a new distinguished color c_r (as a copy of its original color). Then the parent-child relation in T_0 can be easily interpreted based on the unique path to a vertex colored c_r . Let all the colors used in I_1 be $C = \{c_1, \dots, c_m\}$ where m depends on γ (including our distinguished root colors). We construct a tree T from T_0 by attaching i new leaves to every vertex of T_0 of color $c_i \in C$.

We now straightforwardly define an MSO_1 interpretation $I_2 = (\nu_2(x), \eta_2(x, y), \lambda_2^i)$ for $i \in [m]$ of rooted C -colored trees in the class of ordinary trees:

- $\nu_2(x)$ simply asserts that x is not a leaf (x has more than one neighbor),
- $\eta_2(x, y) \equiv \text{edge}(x, y)$ (note that T_0 is an induced subgraph of T), and
- $\lambda_2^i(x)$ asserts that x has precisely i neighbors which are leaves—this has a routine brute-force expression by existential quantification of the i leaf neighbors, coupled by non-existence of $i + 1$ leaf neighbors.

Clearly, $T^{I_2} \simeq T_0$ (including the colors of T_0). We finally set $I = I_1 \circ I_2$ (interpretability is a transitive concept) and $\varphi = \psi^I$ and we are done. \square

With Lemma 4.14 at hand, it is now easy to derive Corollary 4.13 from previous Theorem 4.7 applied to the tree T .

Proof of Corollary 4.13: We invoke Lemma 4.14 to construct the tree T and formula φ , and then apply Theorem 4.7 to them (for a tree, $tw(T) = \tau = 1$, but note that φ now depends on both ψ and γ). Since $S_\psi(G)$ is essentially identical to $S_\varphi(T)$, up to the coordinates corresponding to $V(T) \setminus V(G)$ which are all zero by Lemma 4.14, the solution to the shifted problem with $S_\psi(G)$ is the same as the computed solution to the shifted problem with $S_\varphi(T)$. \square

Finally, we add a small remark regarding the input G in Corollary 4.13; we are for simplicity assuming that G comes along with its γ -expression since it is currently not known how to efficiently construct a γ -expression for an input graph of fixed clique-width γ . Though, one may instead use the result of [12] which constructs in FPT a so-called rank-decomposition of G which can be used as an approximation of a γ -expression for G (with up to an exponential jump, but this does not matter for a fixed parameter γ in theory).

5 MSO-definable sets: $\text{W}[1]$ -hardness

Recall that natural hard graph problems such as CHROMATIC NUMBER are instances of MSO-PARTITIONING and so also instances of shifted combinatorial optimization. While we have shown an XP algorithm for SCO with MSO-definable sets on graphs of bounded treewidth and clique-width in Theorem 4.7 and Corollary 4.13, it is a natural question whether an FPT algorithm could exist for this problem, perhaps under a more restrictive width measure.

Here we give a strong negative answer to this question. First, we note the result of Fomin et al [7] proving $W[1]$ -hardness of CHROMATIC NUMBER parameterized by the clique-width of the input graph. This immediately implies that an FPT algorithm in Corollary 4.13 would be very unlikely (cf. Section 2). Although, CHROMATIC NUMBER is special in the sense that it is solvable in FPT when parameterized by the treewidth of the input. Here we prove that it is not the case of MSO-PARTITIONING problems and SCO in general, even when considering restricted MSO_1 formulas and shifted \mathbf{c} , and parameterizing by a much more restrictive *treedepth* parameter.

Definition 5.1 (Treedepth). *Let the height of a rooted tree or forest be the maximum root-to-leaf distance in it. The closure $cl(F)$ of a rooted forest F is the graph obtained from F by making every vertex adjacent to all of its ancestors. The treedepth $td(G)$ of a graph G is one more than the minimum height of a forest F such that $G \subseteq cl(F)$.*

Note that always $td(G) \geq tw(G) + 1$ since we can use the vertex sets of the root-to-leaf paths of the forest F (from Definition 5.1) in a proper order as the bags of a tree-decomposition of G .

Theorem 5.2. *There exists a graph FO formula $\varphi(X)$ with a free set variable X , such that the instance of the MSO-PARTITIONING problem given by φ , is $W[1]$ -hard when parameterized by the treedepth of an input simple graph G . Consequently, the shifted problem with $S_\varphi(G)$ is also $W[1]$ -hard (for suitable \mathbf{c}) when parameterized by the treedepth of G .*

We are going to prove Theorem 5.2 by a reduction from $W[1]$ -hardness of CHROMATIC NUMBER with respect to clique-width [7]. As an intermediate step for our purpose, [9] prove that the graphs constructed for the reduction in [7], can be interpreted in a special way (formal details to follow) into labeled rooted trees of height 5, where the parameter is the number of labels. We, in turn, prove here that these labels can be traded for increased height of a tree and certain additional edges belonging to the tree closure. Consequently, the property of a set X of vertices to be independent in the original graph can now be expressed by a certain fixed formula $\varphi(X)$ (independent of the parameter) over a plain simple graph which is of bounded treedepth. So the MSO-PARTITIONING instance given by φ is indeed $W[1]$ -hard when parameterized by the treedepth.

We start with formulating the needed special reformulation of the aforementioned result of Fomin et al [7] on hardness of CHROMATIC NUMBER.

Definition 5.3 (Tree-model [10]). *We say that a graph G has a tree-model of m labels and depth d if there exists a rooted tree T such that*

1. *the set of leaves of T is exactly $V(G)$,*
2. *the length of each root-to-leaf path in T is exactly d ,*
3. *each leaf of T is assigned one of m labels,*
4. *and the existence of a G -edge between $u, v \in V(G)$ depends solely on the labels of u, v and the distance between u, v in T .*

Let $\mathcal{TM}_m(d)$ denote the class of all graphs with a tree-model of m labels and depth d .

Theorem 5.4 ([9]). *The graphs constructed as the “hard” instances of CHROMATIC NUMBER in [7] belong to $\mathcal{TM}_m(5)$ where m is the considered parameter. Consequently, the CHROMATIC NUMBER problem considered on the classes $\mathcal{TM}_m(5)$ is $W[1]$ -hard when parameterized by m .*

Proof of Theorem 5.2: We use a reduction from the instance described in Theorem 5.4. Let $G \in \mathcal{TM}_m(5)$ and r be an input of CHROMATIC NUMBER, i.e., the question is whether G is r -colorable. Let T be a tree-model of m labels and depth 5 (Definition 5.3) of G . We are going to construct a formula φ and a graph H of treedepth at most $5m + 7$ such that, $V(G) \subseteq V(H)$ and $X \subseteq V(G)$ is independent if and only if $H \models \varphi(X)$. Moreover, for $Y \subseteq V(H)$ such that $Y \not\subseteq V(G)$, it must hold $H \models \varphi(Y)$ if and only if $Y = V(H) \setminus V(G)$. Then, clearly, $(H, r + 1)$ will be a YES instance of the MSO-PARTITIONING problem given by φ if, and only if, G is r -colorable. This would be the desired reduction.

The rest of the proof is devoted to the construction of φ and H . Let $M = \{1, \dots, m\}$ be the set of labels from Definition 5.3 and let lab map $V(G)$, the set of leaves of T , into M . There exist graphs L_1, \dots, L_5 (self-loops allowed), each on the vertex set M , such that the following holds for any $u, v \in V(G)$ by Definition 5.3: $uv \in E(G)$ if and only if the least common ancestor of u, v in T is at distance $i \leq 5$ from u and $\{lab(u), lab(v)\} \in E(L_i)$.

A graph H_1 is constructed from T as follows:

- All vertices and edges of T are included in H_1 (the labels from T are ignored), the leaves of T have no label in H_1 while all the non-leaf nodes get a new label τ in H_1 .
- For every non-leaf node $x \in V(T)$ at distance $i \leq 5$ from the leaves of T , a disjoint copy L^x of the graph L_i is created and added to H_1 , such that the vertices of L^x receive the same (new) label λ and x is made adjacent to all vertices of L^x .
- Every leaf z of T , for $i = 1, \dots, 5$, is connected by an edge in H_1 to the copy of the vertex $lab(z)$ in L^x , where x is the ancestor of z at distance i from z .

First of all, it is easy to see that H_1 is of treedepth at most $5m + 6 = 1 + 5(m + 1)$, since H_1 is contained in the closure of a tree obtained from T by “splitting” each non-leaf node x to a path on $m + 1$ vertices forming the set $\{x\} \cup V(L^x)$ (which is of cardinality $m + 1$). Second, we observe that the graph H_1 encodes the edges of G as follows; (*) for $u, v \in V(G)$, we find the least common ancestor x of u and v among the τ -labeled vertices of H_1 , and then we test whether there exist λ -labeled neighbors u', v' of x (and so $u', v' \in V(L^x)$) such that $uu', vv' \in E(H_1)$ and also $u'v' \in E(L^x)$.

Assume for now that (*) the edges of G are encoded in a binary predicate γ , such that $uv \in E(G) \iff H_1 \models \gamma(u, v)$. With γ , we can easily define a desired

formula φ_1 such that $H_1 \models \varphi_1(X)$ if, and only if, $X \subseteq V(G)$ is independent in G or $X = V(H_1) \setminus V(G)$. It is

$$\begin{aligned} \varphi_1(X) \equiv & \left[\forall x \in X (\neg \tau(x) \wedge \neg \lambda(x)) \wedge \forall x, y \in X (x = y \vee \neg \gamma(x, y)) \right] \\ & \vee \forall x ((\tau(x) \vee \lambda(x)) \longleftrightarrow x \in X). \end{aligned}$$

Note that γ does not depend on the original m labels of T .

The remaining two tasks are; to express γ in FO over H_1 , and to “get rid of” possible self-loops and the labels τ, λ in H_1 by transforming φ_1 over H_1 into equivalent φ over simple unlabeled H . We finish these tasks as follows.

We recursively define $\alpha_0(x, y) \equiv (x = y)$ and, for $i = 1, \dots, 5$, $\alpha_i(x, y) \equiv \tau(y) \wedge \exists z (\text{edge}(z, y) \wedge \alpha_{i-1}(x, z))$. The meaning of $\alpha_i(x, y)$ is that y is an internal node of T at distance i from x (where x will be a leaf of T but this is not enforced by α_i). The above encoding $(*)$ of the edges of G into H_1 can now be literally expressed as

$$\begin{aligned} \gamma(u, v) \equiv & \exists x \left[\tau(x) \wedge \left(\bigvee_{i=1}^5 \alpha_i(u, x) \wedge \alpha_i(v, x) \wedge \neg \exists x' (\alpha_{i-1}(u, x') \wedge \alpha_{i-1}(v, x')) \right) \right. \\ & \wedge \exists u', v' (\lambda(u') \wedge \lambda(v') \wedge \text{edge}(x, u') \wedge \text{edge}(x, v') \\ & \left. \wedge \text{edge}(u, u') \wedge \text{edge}(v, v') \wedge \text{edge}(u', v') \right), \end{aligned}$$

which is an FO formula independent of H_1 and given T .

Lastly, we observe that H_1 has no vertices of degree 1. We hence construct H from H_1 by adding one new degree-1 neighbor to every τ -labeled vertex of H_1 , adding two new degree-1 neighbors to every λ -labeled vertex of H_1 without self-loop, adding three new degree-1 neighbors to every λ -labeled vertex of H_1 with self-loop, and removing all the loops. The resulting simple graph H is of treedepth at most $5m + 7$ (in fact, again $\leq 5m + 6$), and one can identify the original vertices of H_1 as those having degree > 1 in H . The labels τ, λ and the self-loops of H_1 (as used in the formula φ_1) can be routinely interpreted by FO formulas, e.g., $\tau(x) \equiv \neg \delta_1(x) \wedge \exists y (\delta_1(y) \wedge \text{edge}(x, y)) \wedge \forall y, y' [(\delta_1(y) \wedge \text{edge}(x, y) \wedge \delta_1(y') \wedge \text{edge}(x, y')) \rightarrow y = y']$, where $\delta_1(y) \equiv \forall z, z' [(\text{edge}(y, z) \wedge \text{edge}(y, z')) \rightarrow z = z']$. Such an interpretation defines desired φ from φ_1 . \square

6 Conclusions and open problems

We close with several open problems we consider interesting and promising.

Parameterizing by r . It is interesting to consider taking r as a parameter. For example, Fluschnik et al. [6] prove that the MINIMUM SHARED EDGES problem is FPT parameterized by the number of paths. Omran et al. [21] prove that the MINIMUM VULNERABILITY problem is in XP with the same parameter. Since both problems are particular cases of the shifted problem, we ask whether the shifted problem with S being the set of $s - t$ paths of a (di)graph lies in XP or is NP-hard already for some constant r .

Further uses of Lemma 4.3. For example, which interesting combinatorial sets S can be represented as n -fold integer programs [22,19] such that the corresponding polyhedra are decomposable?

Approximation. The MINIMUM VULNERABILITY problem has also been studied from the perspective of approximation algorithms [21]. What can be said about the approximation of the shifted problem?

Going beyond 0/1. The results in Section 3 are the only known ones in which S does not have to be 0/1. What can be said about the shifted problem with such sets S that are not given explicitly, e.g., when S is given by a totally unimodular system?

References

1. Sepehr Assadi, Ehsan Emamjomeh-Zadeh, Ashkan Norouzi-Fard, Sadra Yazdanbod, and Hamid Zarrabi-Zadeh. The minimum vulnerability problem. *Algorithmica*, 70(4):718–731, 2014.
2. Stephen Baum and Leslie E. Trotter Jr. Integer rounding and polyhedral decomposition for totally unimodular systems. In *Optimization and Operations Research*, pages 15–23. Springer, 1978.
3. Robert Brederbeck, Piotr Faliszewski, Rolf Niedermeier, Piotr Skowron, and Nimrod Talmon. Elections with few candidates: Prices, weights, and covering problems. In *Proc. ADT 2015*, volume 9346 of *LNCS*, pages 414–431, 2015.
4. Jianer Chen, Xiuzhen Huang, Iyad A. Kanj, and Ge Xia. Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.*, 72(8):1346–1367, 2006.
5. Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.
6. Till Fluschnik, Stefan Kratsch, Rolf Niedermeier, and Manuel Sorge. The parameterized complexity of the minimum shared edges problem. In *FSTTCS’15*, volume 45; 45 of *LIPIcs*, pages 448–462. Schloss Dagstuhl, 2015.
7. Fedor Fomin, Petr Golovach, Daniel Lokshtanov, and Saket Saurab. Clique-width: On the price of generality. In *SODA’09*, pages 825–834. SIAM, 2009.
8. Eugene C. Freuder. Complexity of K -tree structured constraint satisfaction problems. In *Proc. of the 8th National Conference on Artificial Intelligence*, pages 4–9, 1990.
9. Jakub Gajarský, Michael Lampis, and Sebastian Ordyniak. Parameterized algorithms for modular-width. In *IPEC’13*, volume 8246, pages 163–176, 2013.
10. Robert Ganian, Petr Hliněný, Jaroslav Nešetřil, Jan Obdržálek, Patrice Ossona de Mendez, and Reshma Ramadurai. When trees grow low: Shrubs and fast MSO_1 . In *MFCS’12*, volume 7464, pages 419–430. Springer, 2012.
11. Dion Gijswijt. Integer decomposition for polyhedra defined by nearly totally unimodular matrices. *SIAM Journal on Discrete Mathematics*, 19(3):798–806, 2005.
12. Petr Hliněný and Sang-il Oum. Finding branch-decompositions and rank-decompositions. *SIAM J. Comput.*, 38(3):1012–1032, 2008.
13. Dorit S. Hochbaum and J. George Shanthikumar. Convex separable optimization is not much harder than linear optimization. *J. ACM*, 37(4):843–862, October 1990.

14. Volker Kaibel, Shmuel Onn, and Pauline Sarrazzolles. The unimodular intersection problem. *Oper. Res. Lett.*, 43(6):592–594, 2015.
15. Dušan Knop, Martin Koutecký, Tomáš Masařík, and Tomáš Toufar. Simplified algorithmic metatheorems beyond MSO: Treewidth and neighborhood diversity, 2016.
16. Petr Kolman, Martin Koutecký, and Hans Raj Tiwary. Extension complexity, MSO logic, and treewidth, July 13 2016.
17. Stephan Kreutzer. Algorithmic meta-theorems. *Electronic Colloquium on Computational Complexity (ECCC)*, 16:147, 2009.
18. Asaf Levin and Shmuel Onn. Shifted matroid optimization. *Oper. Res. Lett.*, 44:535–539, 2016.
19. Jesús A. De Loera, Raymond Hemmecke, and Matthias Köppe. *Algebraic and Geometric Ideas in the Theory of Discrete Optimization*, volume 14 of *MOS-SIAM Series on Optimization*. SIAM, 2013.
20. Timm Oertel, Christian Wagner, and Robert Weismantel. Integer convex minimization by mixed integer linear optimization. *Oper. Res. Lett.*, 42(6-7):424–428, 2014.
21. Masoud T. Omran, Jörg-Rüdiger Sack, and Hamid Zarrabi-Zadeh. Finding paths with minimum shared edges. *J. Comb. Optim.*, 26(4):709–722, 2013.
22. Shmuel Onn. *Nonlinear Discrete Optimization*. Zurich Lectures in Advanced Mathematics. European Mathematical Society, 2010. available online at: <http://ie.technion.ac.il/~onn/Book/ND0.pdf>.
23. Michaël Rao. MSOL partitioning problems on graphs of bounded treewidth and clique-width. *Theor. Comput. Sci.*, 377(1-3):260–267, 2007.
24. Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume 24 of *Algorithms and Combinatorics*. Springer, 2003.
25. Giacomo Zambelli. Colorings of k-balanced matrices and integer decomposition property of related polyhedra. *Oper. Res. Lett.*, 35(3):353–356, 2007.
26. Günter M. Ziegler. *Lectures on polytopes*, volume 152 of *Graduate Texts in Mathematics*. Springer-Verlag, 1995.